

10 Questions about Boggle Logic Puzzles

Jonathan Needleman

May 21, 2015

Abstract

Boggle logic puzzles are based on the popular word game Boggle, where you are given list of words, and your goal is to recreate a Boggle board. In this paper we give an overview of known results and then propose a number of problems related to these puzzles.

1 Introduction

Boggle (©Hasboro inc) is a popular word search game where players compete to find as many words as they can in a 4×4 grid of letters [3]. Boggle logic puzzles on the other hand is the game of Boggle played in reverse. A list of words are given and you need to recreate the board. This paper both summarizes the known results and raises 10 open problems related to these puzzles, many of which are suitable for undergraduate research. However, before we explore these questions, we first give an overview the rules of Boggle.

For a word to be valid it must be at least three letters long, and consecutive letters in the word must be adjacent on the board, either horizontally, vertically or diagonally. Furthermore, each box on the board cannot be used more than once in forming a word. For instance the word MOMATH is on the board because of the path in figure 1. However, BIBS is not on the board, because

G	O	M	C
M	T	I	B
H	A	N	S
U	E	A	W

Figure 1: A Boggle board

there is only one “B” on the board, and HUG is not on the board because the “U” and “G” are not adjacent.

A Boggle logic puzzle is a list of words that can be found in a unique board (up to rotation and reflection). To get a feel for this try the following puzzle:

act, ape, ate, cop, end, old

The goal is to fill in a 3×3 board so that each of the six words appear according to the rules of Boggle. In 2007 Mark Zegarelli published a book [6] of Boggle logic puzzles. All of his puzzles are on a full 4×4 board, but they all have at least one letter filled in. This is most likely to guarantee uniqueness (there are no symmetries in his puzzles).

This article is an attempt to summarize what is known about the mathematics of Boggle logic puzzles, and then give the reader a number of open questions that touch upon a wide variety of fields. However, to do this, we need to translate the problem into mathematical language. We will also make some assumptions to simplify the initial problems.

2 The mathematical language of Boggle

2.1 Basic notation

In order to make the problems more tractable, we assume every letter on a given board is unique. This will prevent us from having to look at many different cases depending on which letters repeat and where they are placed. In section 4.3 we will address some questions that pertain to puzzles with repeated letters.

For an $n \times n$ board, instead of using letters, we will use the numbers $1 \dots n^2$. By doing so we ignore English spelling of words, and let any sequence of adjacent, non-repeating numbers be a word. We denote words with hyphens between “letters” such as $2 - 4 - 5$. The set of all $n \times n$ boards of this form are denoted by \mathcal{B}_n .

To this point most research has focused on 3×3 boards, so we fix a standard 3×3 board **S** to be as in figure 2a

1	2	3
8	9	4
7	6	5

(a)

2	1	8
9	3	4
7	6	5

(b)

Figure 2: The standard board **S**, and a permutation.

It is necessary to talk about other boards besides **S**. To do this we use **S** and permutations. Let S_9 be permutations of the numbers $[1 \dots 9]$. These permutations to refer to other boards in the following way: if $g \in S_9$ then we let $[g]$ be the board $g\mathbf{S}$. For example, the element $(12)(359) \in S_9$ gives the board $[(12)(359)]$ seen in figure 2b by switching the 1 and the 2, and also replacing the 3 with a 5, the 5 with a 9 and a 9 with a 3.

In order to reference words, let $W(\mathbf{B})$ be the set of all words in the board \mathbf{B} that are at least two letters long. Since there is no way to distinguish two different boards that contain exactly the same words (such as a reflection of the board), we say two boards $\mathbf{B}, \mathbf{B}' \in \mathcal{B}_n$ are equivalent if and only if $W(\mathbf{B}) = W(\mathbf{B}')$, and write this as $\mathbf{B} \cong \mathbf{B}'$. It turns out $\mathbf{B} \cong \mathbf{B}'$ if and only if the boards are a rotation or a reflection of each other.

This terminology allows us to mathematically define Boggle logic puzzles.

Definition 1. *An $n \times n$ **Boggle logic puzzle** is a list of words P so that there is $\mathbf{B} \in \mathcal{B}_n$ with $P \subset W(\mathbf{B})$ and whenever there is another $\mathbf{B}' \in \mathcal{B}_n$ with $P \subset W(\mathbf{B}')$ then $\mathbf{B} \cong \mathbf{B}'$.*

All this is saying is that the list of words (or puzzle) P can be found in one and only one board up to equivalence.

2.2 Boggle and graphs

While it is good to have a formal definition of these puzzles, we still need a language in which to prove various properties of these puzzles. Graph theory provides a great a language to do this. This is because Boggle logic puzzles are entirely about which letters are adjacent to each other, and graph theory is the mathematics of adjacency.

We can view the Boggle board as a graph by thinking of boxes on the board as vertices, with edges placed between adjacent boxes. For a $n \times n$ board we call this graph \mathcal{K}_n because it is the well known $n \times n$ king's graph. Furthermore, we can view \mathcal{K}_n as a labeled graph by labeling the vertices with the letters of the standard board. For example, the labeled \mathcal{K}_3 is shown in figure 3. In the future we will suppress the labels to prevent clutter.

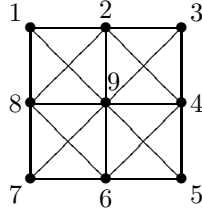


Figure 3: Labeled \mathcal{K}_3

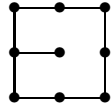
In addition to viewing the board as a graph, we can also interpret the puzzles themselves as graphs. For a standard list of words $P \subset W(\mathbf{S})$ we view the “letters” in the words of P as vertices, and place edges between them if the letters are adjacent in some word of P . This graph $\mathcal{G}(P)$ is called the **adjacency graph** of P . For instance if $P = \{9-1-2, 2-3-4-5, 5-6-7-9, 7-9-8\}$ then $\mathcal{G}(P)$ would be as in figure 4a. The vertices are placed in the same relative position as in the labeled \mathcal{K}_3 graph for consistency.

The language of graph theory allows us to determine when two puzzles $P_1, P_2 \subset W(\mathbf{B})$ are really the same puzzle. If $\mathcal{G}(P_1)$ and $\mathcal{G}(P_2)$ are isomorphic as graphs then we say P_1 is **equivalent** to P_2 and write $P_1 \cong P_2$. In non-technical language this is saying both puzzles P_1 and P_2 provide the exact same information about which letters are adjacent to others.

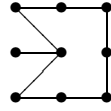
Solving a Boggle logic puzzle P becomes an exercise of figuring out how to view an adjacency graph $\mathcal{G}(P)$ as a subgraph of \mathcal{K}_n . If you can do this, the labeling from $\mathcal{G}(P)$ can then be used to solve the puzzle. With this in mind, we introduce what we believe to be a novel definition in graph theory.

Definition 2. Let \mathcal{H} be a subgraph of \mathcal{G} . Then \mathcal{H} is a **labeling subgraph** of \mathcal{G} if whenever any subgraph \mathcal{H}' of \mathcal{G} is isomorphic to \mathcal{H} by ϕ , then the isomorphism ϕ induces an automorphism on \mathcal{G} .

This definition is rather technical, but the idea behind it is that any labeling of \mathcal{H} induces a unique, up to symmetry, labeling of \mathcal{G} . To better understand what this definition means, let us look at a non-example for \mathcal{K}_3 . The graph in figure 4a is not a labeling subgraph of \mathcal{K}_3 . This is because of the graph in figure 4b. These two graphs are isomorphic subgraphs of \mathcal{K}_3 by exchanging the left hand side vertex (labeled 8 in \mathcal{K}_3) with the center vertex (labeled 9 in \mathcal{K}_3). However, this exchange is not an isomorphism of \mathcal{K}_3 since 8 is adjacent to 5 vertices while 9 is adjacent to 8 vertices in \mathcal{K}_3 .



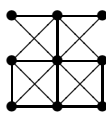
(a) $\mathcal{G}(P)$



(b) A non-labeling subgraph

Figure 4: Non-labeling subgraphs

This definition is designed so that P is a puzzle if and only if $\mathcal{G}(P)$ is a labeling subgraph of \mathcal{K}_n . In 1972 Entringer and Erdős introduced a similar idea called a unique subgraph [1]. The idea is that a subgraph \mathcal{H} of \mathcal{G} is unique if and only if there is no other subgraph isomorphic to \mathcal{H} . For a puzzle P to have a unique solution it may seem $\mathcal{G}(P)$ only needs to be a unique subgraph of \mathcal{K}_n up to automorphism of \mathcal{K}_n . However, this is not strong enough as seen by the graph \mathcal{U} in figure 5a. All subgraphs \mathcal{K}_3 that are isomorphic to \mathcal{U} come from automorphism of \mathcal{K}_3 , that is, they are just rotations and reflections of \mathcal{U} . However, this is not a labeling subgraph because \mathcal{U} is isomorphic to itself by a map that swaps the two upper corners (1 and 3). This map is not an automorphism of \mathcal{K}_3 . In terms of puzzles this means that both the boards in figures 5b and 5c would be solutions to any puzzle P with $\mathcal{G}(P) = \mathcal{U}$.



(a) \mathcal{U}

1	2	3
8	9	4
7	6	5

(b) A solution to \mathcal{U}

3	2	1
8	9	4
7	6	5

(c) Another solution

Figure 5: Unique but non-labeling

3 Known results: The extremes

There are two natural questions concerning the extremal behavior of Boggle logic puzzles. The first is, “What are the fewest words needed to create a puzzle”? The second is “How long does a list of words need to be in order to guarantee that you can uniquely recreate the board”?

For a 3×3 board the first question is answered in [4] with the following theorem:

Theorem 3.1. *Any 3×3 Boggle logic puzzle with no repeated letters and using only three-letter words must contain at least 6 different words.*

This result is readily extended to longer word lengths.

The second question has not been answered as satisfactory, but has the following partial answer is presented in [5].

Theorem 3.2. *For a 3×3 Boggle board with no letters repeated, one needs 137 different (out of 160 possible) three-letter words to guarantee a unique solution. For four-letter words, 377 words are needed out of 496 possible words.*

3.1 Minimal Solutions

The paper [4] shows any 3×3 Boggle logic puzzle must be made of at least 6 three letter words. This paper studies two-letter words instead of looking at three-letter words. In fact, the proof of theorem 3.1 only proves that at least 11 two-letter words are needed to create a puzzle. This is strong enough to claim that at least 6 three-letter words are needed. However, every puzzle found so far uses at least 12 two-letter words. This leads to our first open question.

Open Question 1. *Are there any 3×3 puzzles with nine distinct letters that have only 11 two-letter words?*

Our conjecture is that the answer to this question is no. Some modifications to the proof of theorem 3.1 is probably all that is necessary, making this problem accessible to undergraduates.

The next question, however, is more involved and introduces the idea of minimal puzzles. A **minimal puzzle** P is a puzzle where if you break up a word into two smaller words (possibly a one letter word) then P is no longer

a puzzle. Equivalently, removing an edge from $\mathcal{G}(P)$ yields a graph that is not a labeling subgraph. Essentially, a minimal puzzle is a puzzle where all the information given in the puzzle is needed to solve the puzzle. This begs the following question:

Open Question 2. *How many inequivalent minimal $n \times n$ puzzles are there?*

From a puzzle maker's point of view this is asking about how robust Boggle logic puzzles are as a source of puzzles. Will they be making the same puzzles over and over just with different words? We expect that 3×3 boards are not a great source of puzzles, but 4×4 and larger boards probably will yield a large number of puzzles.

Once question 1 has been answered there is another version of question 2 to answer for 3×3 boards. How many minimal puzzles are there with either 11 or 12 two-letter words (depending on the answer to question 1)? This question has the potential to be easier because all puzzles with either 11 or 12 words would be minimal, so one only needs to check how many words are in the puzzle. For a 3×3 boards there are either $\binom{20}{11}$ or $\binom{20}{12}$ word lists that may be minimal, and so these questions can probably be answered by computer. However a 4×4 board has 42 edges instead of 20 so the number of subgraphs of a fixed number of edges is likely larger than a computer can handle, so it would be ideal to have analytic solutions.

3.2 Maximal Solutions

Another natural question about Boggle logic puzzles comes from the game of Boggle itself. Imagine playing a round of Boggle and at the end of the round you have a long list of words. Is there an easy way to determine if you can recreate the board from your list? The first approach you may take is to see if there is a number N where, if you have that many words, you are guaranteed to be able to recreate the board. Theorem 3.2 from [5] is a such a result for 3×3 boards with three- and four-letter words.

The idea of the proof is relatively simple: find a board \mathbf{B} that maximizes the number of words in common with the standard board \mathbf{S} . One more than the number of common words is the cut-off value N to guarantee a unique solution. For both three- and four-letter words this board is $[(1\ 3)]$ as seen in figure 5c. However, the proof for three-letter words and four-letter are essentially done as independent proofs. This leads to some open questions which should be accessible to undergraduates.

Open Question 3. *Does the board $[(1\ 3)]$ have the most k -letter words in common with \mathbf{S} of any 3×3 board, and if so is there a uniform proof?*

The 3×3 board appears to have some degeneracy that does not appear in larger boards. In particular, the board $[(1\ 3)]$, which swaps adjacent corners, is special in the 3×3 case due to the fact that both 1 and 3 are both adjacent to a common side letter 2. Since this is not the case for $n \times n$ boards with $n \geq 4$

swapping adjacent corners probably will not maximize common words. Instead we provide the following conjecture in the form of an open question.

Open Question 4. *Given an $n \times n$ board \mathbf{B} with $n \geq 4$, does the board which comes from swapping a corner letter with an adjacent side letter maximize the number of k -letter words in common with \mathbf{B} ?*

4 Open Questions

In this section we introduce a number of open questions unrelated to any known result.

4.1 Probabilistic Questions

In section 3.2 we saw that it takes a large number of words to guarantee a unique solution, probably more than one could expect to find in any actual game of Boggle. Instead one can ask probabilistic questions in hopes of obtaining answers that may be more in line with what one may see in a game of Boggle. With this in mind we ask two questions, one about the mean, and one about the median.

Open Question 5. *For a given $n \times n$ board what is the expected number of k -letter words needed to recreate the board?*

Open Question 6. *For a given $n \times n$ board what is the number of k -letter words needed so that there is a 50% chance the board can be recreated?*

The naive brute force methods to answer these questions quickly run into problems. Let us just think about these problems in the simplest case, a 3×3 board \mathbf{B} and two-letter words.

For question 5, one approach is to take a random word $w_1 \in W(\mathbf{B})$, and determine the average number of boards that contain w_1 . Then choose another word w_2 at random and determine how many boards are expected to include both w_1 and w_2 . Continue adding words until you expect only one board. For one word this computation is doable, but once you add more words the dependence structure becomes quite complicated. The number of boards containing words w_1, \dots, w_m depends greatly on the adjacency structure of the words chosen. This makes a direct computation of the mean number of boards containing these words difficult.

On the other hand, an answer to question 6 is likely related to the answer to question 2. If one can determine how many length m lists of two-letter words have a unique solution then one essentially has an answer to question 6 since the total number of lists is relatively easy to count.

4.2 Computational Complexity

It is natural to ask about the computational complexity of solving a Boggle logic puzzle. Up to this point we have been thinking of puzzles on a $n \times n$

boggle board. For a given $n \times n$ puzzle P this amounts to determining how the adjacency graph $\mathcal{G}(P)$ can be viewed as a subgraph of \mathcal{K}_n . Instead of playing on a standard Boggle board the puzzle can actually be done on any graph. From this perspective solving Boggle logic puzzles is related to solving the subgraph isomorphism problem, a classic NP-complete problem.

The subgraph isomorphism problem asks if a graph \mathcal{H} is isomorphic to a subgraph of \mathcal{G} . This problem can be shown to be NP-complete by showing it is equivalent to the clique problem (is a complete graph a subgraph) or by the Hamiltonian cycle problem (does a graph have a Hamiltonian cycle), both of which are known to be NP-complete [2].

Returning to Boggle puzzles on standard boards we observe that the graph \mathcal{K}_n does not contain a complete graph of order 5, and it contains a Hamiltonian cycle for all n . This makes it unclear if solving Boggle logic puzzles is NP-complete, and so we raise the following question.

Open Question 7. *What is the computational complexity of solving $n \times n$ Boggle logic puzzles?*

A related and potentially more difficult question is:

Open Question 8. *What is the computational complexity of creating $n \times n$ Boggle logic puzzles?*

Question 8 is asking “Is a graph \mathcal{H} a labeling subgraph of \mathcal{K}_n ”? For the subgraph isomorphism problem one wants the existence of an isomorphism from \mathcal{H} to a subgraph of \mathcal{K}_n . For the labeling subgraph problem one wants to know there exists *exactly* eight isomorphisms from \mathcal{H} to subgraphs of \mathcal{K}_n . Since this is a counting problem creating Boggle logic puzzles is at least as difficult as solving them.

There is no reason to limit oneself to the computational complexity of solving Boggle logic puzzles instead of asking about labeling subgraphs of general graphs. The only difference is that if \mathcal{H} is a labeling subgraph of \mathcal{G} one needs to show the number of isomorphisms from \mathcal{H} to subgraphs of \mathcal{G} is equal to the number of automorphisms of \mathcal{G} .

4.3 Repeated Letters

To this point we have only discussed problems that assume all the letters on the board are different. Of course, all those questions can be asked again assuming that some letters are allowed to repeat, but as we will see this does not necessarily make sense in all contexts. Instead, we introduce some questions about which boards with repeated letters are worthy of study for Boggle logic puzzles.

Let us look at all 3×3 boards with eight 1s and one 2 for letters. It is not hard to check that every board with these letters must contain the same words. In other words, it does not matter where the 2 is placed. When all letters are different the only equivalence between boards occur from the symmetries of a square, but in this case all boards are equivalent. This means it does not make

sense to study Boggle logic puzzles with these letters since there is no list of words that will distinguish boards.

Before moving on we introduce a way to discuss which letters are being used in a given board. Let $\lambda = (\lambda_1, \dots, \lambda_m)$ be a partition of n^2 . That is $\sum \lambda_i = n^2$ with $\lambda_i \in \mathbb{N}$ and $\lambda_i \geq \lambda_{i+1}$. We will say a $n \times n$ board is type λ if it contains λ_i copies of the letter i . For instance, boards of type $\lambda = (8, 1)$ are all the 3×3 boards with eight 1s, and a single 2. Whereas boards of the form $\lambda = (1, 1, 1, 1, 1, 1, 1, 1)$ (which we will abbreviate as $\lambda(1^9)$) are all boards with nine distinct letters. We then let $\mathcal{B}(\lambda)$ be all boards of type λ .

We have seen boards of type $(8, 1)$ are not interesting to study as Boggle logic puzzles, so the question becomes which boards actually are interesting to study. We will say board \mathbf{B} is **solvable** if all their equivalent boards come from symmetries of a square. We know if all the letters are different this is the case, but this isn't always true when there are repeated letters.

Open Question 9. *Let λ be a partition of n^2 . Which $\mathbf{B} \in \mathcal{B}(\lambda)$ are solvable?*

For $\lambda = (1^{n^2})$ all boards are solvable, but this is not an interesting result because no letters are repeated. Are there any other solvable boards? Yes! In fact all boards of type $(2, 1^7)$ (boards with a single letter repeated once) are solvable. There is no published proof of this fact so we briefly outline the ideas here.

First, boards with two adjacent 1s are inequivalent to boards with non-adjacent 1s. For boards with two adjacent 1s look at how many words can be of the form $1 - x - 1$ and $1 - 1 - x$. This pair of numbers is distinct for each placement of two adjacent 1s. For non-adjacent 1s, checking how many words are of the form $1 - x - 1$ is not quite enough as there are two different boards that have two words of that form, but then those boards can be seen to be inequivalent.

This leads to one final open question.

Open Question 10. *For which λ , a partition of n^2 , are all $\mathbf{B} \in \mathcal{B}(\lambda)$ are solvable?*

5 Conclusions

Boggle logic puzzles are fun puzzles that have rich mathematics at their core. The fact that the computational complexity of these puzzles is related to the subgraph isomorphism problem shows how the puzzle is related to an area of active research. On the other hand, the puzzles give rise to other problems in graph theory that are accessible to people with a wide range of backgrounds.

The study of Boggle logic puzzles is very young, and the problems listed in this article is only intended to be a starting place for further research. We hope that in the next edition of this journal many of these questions will be solved and many more new problems proposed.

References

- [1] Entringer, R. C. and Erdős, P, *On the number of unique subgraphs of a graph*. J. Combinatorial Theory B 13, 1972, 112-115.
- [2] Garey, Michael R. and Johnson, David S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman. 1979.
- [3] Hasbro Incorporated. Boggle Game, *Rule Book*, 1997.
- [4] Needleman, Jonathan. *Boggle Logic Puzzles: Minimal Solution*. College Math Journal, Vol. 44, No. 4 September 2013, 293-299.
- [5] Needleman, Jonathan and Porrino, Lauren. *Boggle Logic Puzzles: Maximal non-solutions*. Accepted, Pi Mu Epsilon.
- [6] Zegarelli, Mark. *Sit and Solve BOGGLE Logic Puzzles* Puzzlewright 2007.